Benchmarking dynamic Bayesian network structure learning algorithms

Ghada Trabelsi, Philippe Leray, Mounir Ben Ayed, Adel Alimi

Benchmarking dynamic Bayesian network structure learning algorithms

Ghada Trabelsi
REsearch Group on Intelligent Machines,
National School of Engineers
(ENIS) of Sfax, University of Sfax, Tunisia
Knowledge and Decision Team,
Laboratoire d'Informatique de Nantes
Atlantique (LINA), Nantes, France
Email: ghada.trabelsi@ieee.org

Philippe Leray
Knowledge and Decision Team,
Laboratoire d'Informatique de Nantes
Atlantique (LINA), Nantes, France
Email: philippe.leray@univ-nantes.fr

Mounir Ben Ayed
Adel M. Alimi
REsearch Group on Intelligent Machines,
National School of Engineers (ENIS) of
Sfax, University of Sfax, Tunisia
Email:mounir.benayed@ieee.org
adel.alimi@ieee.org

***Abstract*—Dynamic Bayesian Networks (DBNs) are probabilistic graphical models dedicated to modeling multivariate time series. Two-time slice BNs (2-TBNs) are the most current type of these models. Static BN structure learning is a well-studied domain. Many approaches have been proposed and the quality of these algorithms has been studied over a range of different standard networks and methods of evaluation. To the best of our knowledge, all studies about DBN structure learning use their own benchmarks and techniques for evaluation. The problem in the dynamic case is that we don't find previous works that provide details about used networks and indicators of comparison. In addition, access to the datasets and the source code is not always possible. In this paper, we propose a novel approach to generate standard DBNs based on tiling and novel technique of evaluation, adapted from the "static" Structural Hamming Distance proposed for Bayesian networks.**

***Index Terms*—Dynamic Bayesian Networks, 2-TBN models, Bayesian Network Tiling, Structural Hamming Distance.**

I. Introduction

During the last two decades there has been an increasing interest in the Bayesian Network (BN) formalism [1], [2]. The success of BN as one of the most complete and consistent formalisms for the acquisition and representation of knowledge and for reasoning from incomplete and/or uncertain data return to the fact that: (1) their mathematical basis is rigorously justified; (2) they deal in an innate way with uncertainty (modeled as a joint probability distribution); (3) they are understandable (graphical representation); and (4) they take advantage of locality both in knowledge representation and during inference.

Learning the graphical part (i.e. the structure) of these models from data is an NP-hard problem [3]. Many studies have been conducted on this subject. The most of these works and their result interpretations use standard networks and common performance indicators such as approximation of the marginal likelihood of the obtained model or comparison of the resulting graph with the original graph given in benchmarking tasks with the help of the Structural Hamming Distance (SHD) proposed by Tsamardinos [4].

Dynamic Bayesian networks (DBNs) are a general and flexible model class for representing complex stochastic temporal processes [5]. Some structure learning algorithms have been proposed, adapting principles already used in "static" BNs. Comparing these algorithms is a difficult task because the evaluation technique and/or the reference networks used change over each article. Evaluation of these algorithms is also often restricted to networks with a small number of variables, at the difference of "static" BNs where structure learning has been studied with large reference networks.

We focus on this paper to present two contributions: (1) an algorithm for generating large 2-TBN networks (which could be used as standard 2-TBN benchmarks) by using tiling approach [6], [7] in the dynamic case; (2) an algorithm for the evaluation of a 2-TBN structure learning algorithm by adapting the SHD measure no more correct with temporal networks. We believe that this work can be a usefull tool for benchmarking any 2-TBN structure learning algorithm in a common framework.

Section 2 provides the background of our work with a brief introduction to the evaluation methods used in BN learning. In section 3, we detail our proposed approach to generate large DBN and metric to evaluate the performance of DBN structure learning algorithms. Section 4 describes our experimental results. Finally in section 5, we presents conclusions and future works.

II. Background

A. Dynamic Bayesian networks and structure learning

A dynamic Bayesian Network (DBN) is a probabilistic graphical model devoted to represent sequential systems [5]. More precisely, a DBN defines the probability distribution of a collection of random variables $\mathbf{X}[t]$ where $\mathbf{X} = \{X_1 \ldots X_n\}$ is the set of variables observed along discrete time t.

In this work, we consider a special class of DBNs, namely 2-time slice Bayesian Networks (2-TBN). A 2-TBN is a DBN which satisfies the Markov property of order 1 $\mathbf{X}[t-1] \perp \mathbf{X}[t+1] \mid \mathbf{X}[t]$. As a consequence, a 2-TBN is described by a pair $(M_0, M_\rightarrow)$.

M_0 (**initial model**) is a BN representing the initial joint distribution of the process $P(\mathbf{X}[t = 0])$ and consisting of a direct acyclic graph (DAG) G_0 containing the variables $\mathbf{X}[t = 0]$ and a set of conditional distributions $P(X_i[t = 0] \mid pa_{G_0}(X_i))$ where $pa_{G_0}(X_i)$ are the parents of variable $X_i[t = 0]$ in G_0;

$M_{\rightarrow}$ (**transition model**) is another BN representing the distribution $P(\mathbf{X}[t + 1] \mid \mathbf{X}[t])$ and consisting of a DAG $G_{\rightarrow}$ containing the variables in $\mathbf{X}[t] \cup \mathbf{X}[t + 1]$ and a set of conditional distributions $P(X_i[t + 1] \mid pa_{G_{\rightarrow}}(X_i))$ where $pa_{G_{\rightarrow}}(X_i)$ are the parents of variable $X_i[t + 1]$ in $G_{\rightarrow}$, parents which can belong to time t or $t + 1$.

Daly et al [8] propose a recent and interesting state of the art about BN and DBN learning. From the important literature about BN structure learning, some reference networks emerged. Let us cite some large networks : GENE [9], LINK [10] or PIGS [11]) available in the web. These reference networks are used in order to generate data which is used by the structure learning algorithm. The learnt BN can be evaluated by common performance indicators such as approximation of the marginal likelihood [12] or comparison of the resulting graph with the reference graph with the help of the Structural Hamming Distance (SHD) proposed by Tsamardinos [4].

Contrary to BN, learning structure for the DBN is more difficult for two reasons: foremost the learning complexity induced by adding the temporal dimension; then the unavailability of standard benchmarks. except for instance some reference nertworks with a small number of variables (less than 10), such as Umbrella and Water [1]. Besides, articles dealing with 2-TBN structure learning never use the indicators to argue about the goodness of their proposition.

B. Generation of large "static" Bayesian networks

As we previously noticed, Learning the graphical part (i.e. the structure) of BN from data is an NP-hard problem. Many studies have been conducted on this subject, leading to three different families of approaches: (1) constraint-based methods, (2) score-based methods, (3) local search methods. The two first approaches are usually validated on benchmark models with small number of variables. These methods will underperform if the number of variables increases. The third approach is able to scale to distributions with more than thousands of variables. As existing BN benchmarks where limited in their number of variables, some researchers proposed generating BNs by controlling their size and/or complexity. [13] presents methods for generation of random BN by generating uniformly distributed samples of directed acyclic graphs. They develop a uniform generation of multi-connected and singly-connected networks for a given number of nodes. Generating a very large BN randomly is not very realistic. In many large applications, the global model can be decomposed in coherent repeated subgraphs. [6] proposed a novel algorithm and software for the generation of arbitrarily large BN (e.g., graphical models representing and joint probability distributions) by tiling smaller real-world known networks (tiles). The complexity of the final model is controlled by two parameters : the number of tiling n and the connectivity parameter c which determines the maximum number of connections between one node and the next tile.

C. Evaluation of BN structure learning algorithms

There are several measures proposed in the literature for evaluation of structure learning algorithms. The most popular metric is the BDeu score [2]. Under certain assumptions it corresponds to the a posteriori probability (after having seen the data) of the learned structure.

[14] notice that using the BDeu score as a metric of reconstruction quality has the following two problems. First, the score corresponds to the a posteriori probability of a network only under certain conditions (e.g., a Dirichlet distribution of the hyperparameters); it is unknown to what degree these assumptions hold in distributions encountered in practice. Second, the score depends on the equivalent sample size and network priors used. Since, typically, the same arbitrary value of this parameter is used both during learning and for scoring the learned network, the metric favors algorithms that use the BDeu score for learning. In fact, the BDeu score does not rely on the structure of the original, gold standard network at all; instead it employs several assumptions to score the networks.

When proposing a new structure learning algorithm, Tsamardinos et al. [4] propose an adaptation of the usual Hamming distance between graphs taking into account the fact that some graphs with different orientations can be statistically indistinguishable. As graphical models of independence, several equivalent graphs will represent the same set of dependence/independence properties. These equivalent graphs (also named Markov or likelihood equivalent graphs) can be summarized by a partially DAG (PDAG). This new structural Hamming distance (SHD) compares the structure of the PDAG of the learned and the original networks as described in algorithm 1 in order to only compare orientations that are really statistically distinguishable.

As mentioned before, structure learning is a difficult task. Some works propose using prior knowledge in order to limit the search space, for instance by declaring some forbidden or required edges in the final graph [15]. By dealing with PDAGs, the previous SHD measure only take into account information from learning data, forgetting that some orientations have been provided by prior knowledge.

D. Evaluation of a model learned from data and prior knowledge

When proposing a first algorithm to determine the PDAG of a given graph, [16] also proposed a way to take into account prior background knowledge. This solution is decomposed into three phases. The first phase consists in determining the PDAG. This step can be resolved by keeping the skeleton of the given DAG, and its V-structures, and then applying recursively a set of three rules R_1, R_2, and R_3 in order to infer all the edge orientations compatible with the initial DAG. The second phase consists in comparing this PDAG with the prior knowledge. If some information are conflicting, the algorithm

[1]http://www.cs.huji.ac.il/ galel/Repository/Datasets/water/water.html

Algorithm 1 Structural Hamming distance algorithm [4]

Require: Learned PDAG H; Original PDAG G
Ensure: SHD value
$SHD = 0$
for all edge E different between H and G **do**
  if E is missing in H **then**
    SHD=SHD+1
  end if
  if E is extra in H **then**
    SHD=SHD+1
  end if
  if E is uncorrectly oriented in H **then**
    SHD=SHD+1
  end if
end for

returns an error. The final step consists in iteratively adding the prior knowledge (edges) not present in the PDAG and applying again the previous recursive orientation rules in order to infer all the new edge orientations induced by the addition of the prior knowledge. Meek demonstrates that another rule R_4 is needed in order to complete the three previous ones. We can notice that [17] proposed an optimized implementation of the first phase (PDAG determination).

III. Contributions

We describe here new tools for benchmarking dynamic Bayesian network structure learning algorithms. The first subsection describes a 2-TBN generation algorithm. The second one presents a novel metric for evaluating performance of 2-TBN structure learning algorithms, dealing with temporal background knowledge.

A. Generation of large 2-TBNs

In order to generate a large 2-TBN model, we propose generating two models M_0 and $M_\rightarrow$ from an initial static benchmark BN M, as described in algorithm 2.

First we use the Tsamardinos's work for generating realistic large bayesian networks by tiling. Our method consists in generating a large initial model M_0 and its conditional probability distribution by tiling n copies of the initial model M. Then we use again tiling to generate $M_\rightarrow$ and the transition probability distribution by tiling 2 copies of M_0.

Complexity of the final 2-TBN can be controlled by changing the number of tiling copies and the intra-connectivity c_i (used for generating M_0) or the temporal connectivity c_t (used for generating temporal edges).

B. Evaluation of DBN generated by data and prior knowledge

As 2-TBNs are defined by two graphs G_0 and $G_\rightarrow$, we propose evaluating the structural difference between one theoretical 2-TBN and a learned one by the pair of the structural Hamming distance for the corresponding initial and transition graphs as described in algorithm 3.

As we have seen before, taking into account Markov equivalence by comparing PDAGs is important for BN structure learning evaluation, but it's not sufficient for DBNs. Some temporal information (a priori knowledge) is used for 2-TBN structure learning and can be lost by reasoning with PDAGs.

Algorithm 2 Generation of large 2-TBNs algorithm

Require: BN DAG M, number of copies n, intra-connectivity c_i, temporal connectivity c_t
Ensure: Return initial M_0 and transition models $M_\rightarrow$
M_0 = bn_tiling(M,n,c_i)
$M_\rightarrow$ = bn_tiling(M_0,2,c_t)

Algorithm 3 Structural Hamming distance for 2-TBNs algorithm

Require: Learned PDAG H_0; Learned PDAG $H_\rightarrow$; Original PDAG G_0; Original PDAG $G_\rightarrow$
Ensure: SHD values for initial and transition graphs
1: H_k=$H_\rightarrow$
2: G_k=$G_\rightarrow$
% calculate SHD_0
3: SHD_0=SHD(H_0,G_0)
% Temporal correction for $G_\rightarrow$
4: Select randomly a temporal edge from $G_\rightarrow$
5: Orient this temporal undirected edge in G_k
6: Recursively apply the Meek rules in G_k
7: If there exist any unprocessed temporal edge then repeat 4, 5, 6.
% Temporal correction for $H_\rightarrow$
8: Select randomly a temporal edge from $H_\rightarrow$
9: Orient this temporal undirected edge in H_k
10: Recursively apply the Meek rules in H_k
11: If there exist any unprocessed temporal edge then repeat 8, 9, 10.
% Calculate $SHD_\rightarrow$
12: $SHD_\rightarrow$=SHD(H_k,G_k)
% calculate SHD in 2-TBN
13: SHD = (SHD_0,$SHD_\rightarrow$)

In the case of 2-TBN, two different models are learnt. The first one M_0 doesn't model temporal information, so the usual Tsamardinos' SHD can be used.

The second model named represents the dependency relations between nodes of the same slice t or between the nodes of slices t and $t+1$. We have here an important background (temporal) knowledge, edges between time slices are directed from t to t+1. We then propose to adapt the Tsamardinos' SHD in order to deal with this additional knowledge as proposed in section II-D for BNs. One temporal correction is applied for each PDAG in order to obtain a corrected $PDAG_k$ compatible with the prior knowledge. The structural Hamming distance is then computed between these $PDAG_k$

IV. Validation

A. 2-TBN Benchmark generation

For our implementation, we decided that the initial network M is provided in Hugin[2] format readable by several BN software such as Genie/Smile[3] or Matlab toolboxes such as Causal Explorer[4].

As Causal explorer also proposes an implementation of bn_tiling(), we implemented our 2-TBN benchmark generation in Matlab using these functions, and added another function in order to export an unrolled 2-TBN model in Hugin format. This exported 2-TBN can then be used by several software for data generation and structure learning.

[2]http://www.hugin.com/
[3]http://genie.sis.pitt.edu/
[4]http://www.dsl-lab.org/causal_explorer/index.html

Figure 1 illustrate our algorithm with ASIA [18] generating network, tiled 3 times for the initial model, with a maximum connectivity equal to 3. As we can see, we are now able to generate realistic 2-TBNs with very large domains by choosing any static and well-known benchmarks and controlling the complexity by increasing the number of tiling.

B. Structural Hamming distance for 2-TBN

In Figure 2, we show the interest of the temporal correction proposed for SHD in section III-B. We can notice that the PDAG corresponding to each 2-TBN can lose some temporal information by un-orienting some temporal edges (resp. 1 and 2 for 2-TBN$_0$ and 2-TBN$_1$).

Applying the structural Hamming distance without correction gives us a distance equal to 2 between transition graphs related to 2-TBN$_0$ and 2-TBN$_1$. This distance take into account the missing edge between C_{t+1} and D_{t+1} and the missing orientation of the temporal edge between D_t and D_{t+1}. This distance increases to 5 between 2-TBN$_0$ and 2-TBN$_2$ because of the 2 modifications (one missing edge and one added), but also the 3 missing orientations in 2-TBN$_0$.

Application of our temporal correction orients the temporal edges in the corrected PDAG, but also orients 2 more edges in the transition graph related to 2-TBN$_0$and 2-TBN$_1$.

Applying the structural Hamming distance with correction gives us a distance equal to 1 between transition graphs related to 2-TBN$_0$ and 2-TBN$_1$, which correspond to the "true" missing edge, and a distance equal to 2 for the last model, which also corresponds to the "true" differences.

As we can see in these toy examples, our SHD with temporal correction is better in term of structural comparison of dynamic bayesian networks. The improvement is given by the integration of knowledge (temporal knowledge) in our metric.

V. Conclusion and perspectives

We focus in this paper on providing tools for benchmarking dynamic Bayesian network structure learning algorithms. Our first contribution is a 2-TBN generation algorithm inspired from the Tiling technique proposed by [6]. Our algorithm is able to generate large and realistic 2-TBNs which can then be used for sampling datasets. These datasets can then feed any 2-TBN structure learning algorithm.

Our second contribution is a novel metric for evaluating performance of these structure learning algorithms, by correcting the Structural Hamming distance proposed by [4] in order to take into account temporal background information.

Our next step in this direction is proposing one website by providing some 2-TBNs benchmarks (graphs and datasets) in order to provide common evaluation tools for every researcher interested in 2-TBN structure learning.

Another immediate perspective is the adaptation of the structural Hamming distance in order to take into account any background knowledge (forbidden edges, required ones, partial ordering, ...).

Acknowledgment

The first, third and fourth authors would like to acknowledge the financial support provided by grants from the General Department of Scientific Research and Technological Renovation (DGRST), Tunisia, under the ARUB program 01/UR/11/02.

References

[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo: Morgan Kaufmann Publishers, 1988.

[2] D. Heckerman, D. Geiger, and D. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, vol. 20, pp. 197–243, 1995.

[3] D. Chickering, D. Geiger, and D. Heckerman, "Learning bayesian networks is NP-hard," Microsoft Research Technical Report, Tech. Rep. MSR-TR-94-17, 1994. [Online]. Available: http://citeseer.nj.nec.com/140425.html

[4] I. Tsamardinos, L. E. Brown, and C. F. Constantin, F. Aliferis, "The max-min hill-climbing bayesian network structure learning algorithm," *Mach. Learn.*, vol. 65, no. 1, pp. 31–78, 2006.

[5] K. Murphy, "Dynamic bayesian networks: Representation, inference and learning," PhD, University of California, Berkeley, 2002.

[6] A. R. Statnikov, I. Tsamardinos, and C. Aliferis, "An algorithm for generation of large bayesian networks," Department of Biomedical Informatics, Discovery Systems Laboratory, Vanderbilt University, Tech. Rep. Technical Report DSL-03-01, 2003.

[7] I. Tsamardinos, A. Statnikov, L. E. Brown, and F. A. Constantin, "Generating realistic large bayesian networks by tiling," in *Proceedings of the 19th International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, 2006, pp. 592–597.

[8] R. Daly, Q. Shen, and S. Aitken, "Learning bayesian networks: approaches and issues," *Knowledge Engineering Review*, vol. 26, no. 2, pp. 99–157, 2011.

[9] P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization," *Mol Biol Cell*, vol. 9, p. 3273Ű3297, 1998.

[10] C. Jensen and A. Kong, "Blocking Gibbs sampling for linkage analysis in large pedigrees with many loops," Department of Computer Science, Aalborg University, Denmark, Research Report R-96-2048, 1996.

[11] C. Jensen, "Blocking gibbs sampling for inference in large and complex bayesian networks with applications in genetics," PhD thesis, Aalborg University, Denmark, 1997.

[12] D. Chickering and D. Heckerman, "Efficient Approximation for the Marginal Likelihood of Incomplete Data given a Bayesian Network," in *UAI'96*. Morgan Kaufmann, 1996, pp. 158–168. [Online]. Available: http://research.microsoft.com/ dmax/publications/ml97.pdf

[13] J. S. Ide and F. G. Cozman, "Random generation of bayesian networks," in *Brazilian Symp. on Artificial Intelligence*. Springer-Verlag, 2002, pp. 366–375.

[14] M. Gasse, A. Aussem, and H. Elghazel, "An experimental comparison of hybrid algorithms for bayesian network structure learning," *Machine Learning and Knowledge Discovery*, vol. 7523, pp. 58–73, 2012.

[15] L. M. de Campos and J. G. Castellano, "Bayesian network learning algorithms using structural restrictions," *International Journal of Approximate Reasoning*, pp. 233–254, 2007.

[16] C. Meek, "Causal inference and causal explanation with background knowledge," in *Proceedings of the Eleventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-95)*. San Francisco, CA: Morgan Kaufmann, 1995, pp. 403–410.

[17] D. Chickering, "Learning equivalence classes of bayesian-network structures," *Journal of Machine Learning Research*, vol. 2, pp. 445–498, 2002.

[18] S. Lauritzen and D. Speigelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *Royal statistical Society B*, vol. 50, pp. 157–224, 1988.

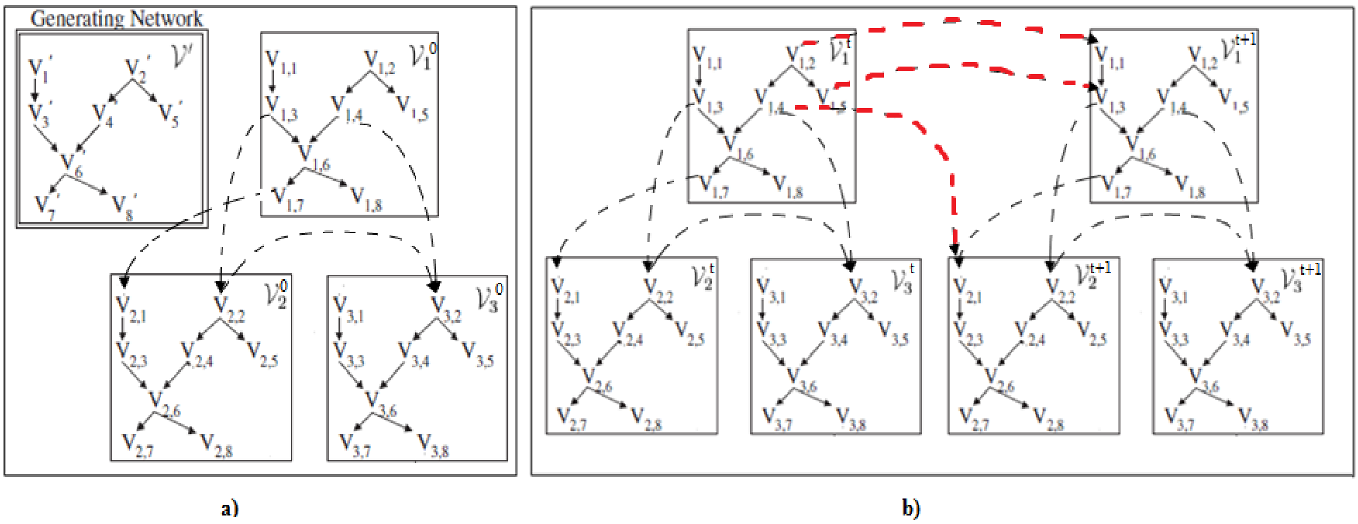

Figure 1. An example output of the 2-TBN generation algorithm. The generating network is ASIA benchmark [18] shown in the top left of figure a). a) The output of initial network consists of three tiles of Asia with the addition of several intraconnecting edges shown with the dashed edges. b) The output of transition network with the addition of several interconnecting edges shown with the dashed red edges.

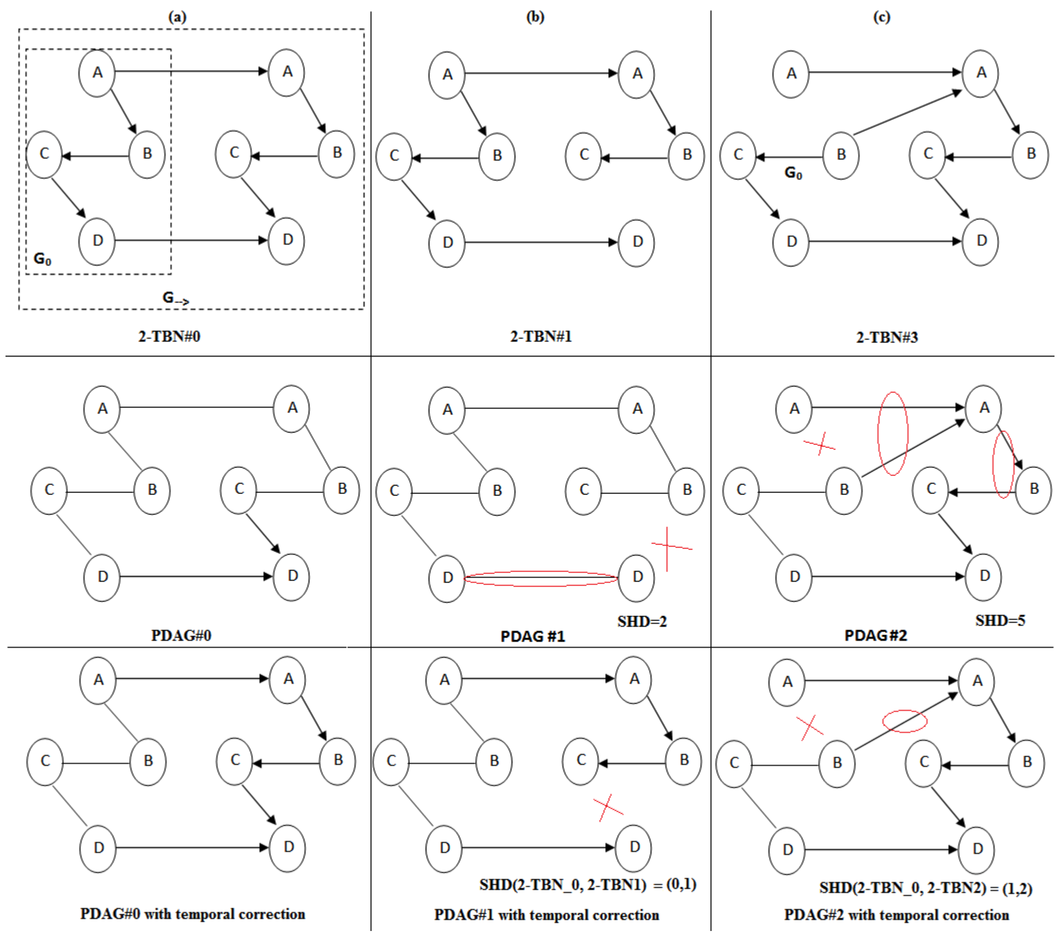

Figure 2. Two examples of structural Hamming distance with or without temporal correction. A first 2-TBN and its corresponding PDAG and corrected PDAG_k are shown in (a). (b) and (c) show two other 2-TBN and their corresponding PDAGs, and the structural Hamming distance with the first model